

Dynamical System-based Robot Reaching Motions by Para-Model Control Approach - A preliminary study -

Loïc Michel

October 4, 2016

Abstract

In this report, we apply the proposed "para-model" framework in order to control the trajectory of a dynamical system-based robot. The optimization of the dynamical performances in closed-loop is performed using a derivative-free optimization algorithm.

1 Model-free control approach

The model-free control methodology has been originally proposed by Fliess & Join [1], which is referred to as a self-tuning controller in [2] and which has been widely and successfully applied to many mechanical and electrical processes. This control law has been designed to "robustify" *a priori* any "unknown" dynamical system for which not only uncertainties and unexpected modifications of the model parameter(s) are considered, but also switched models and models with time-delay(s)...

The principle of this control law consists in building an ultra-local model of the controlled process from the measurements of the input and output signals, but the main disadvantage is that the derivative of the output signal is required. This ultra-local model is a part of an "auto-adjusting" or "extended" PI control and the performances are really good taking into account that no explicit model is *a priori* given - the control is only based on input & output signals.

One of the last contributions, called *para-model agent* (PMA) [3], removes the use of the derivatives and replaces them by an initialization function. This contribution can be considered as a *derivative-free & model-free control scheme*. The last application, which has been successfully experimentally validated, deals with the nonlinear control of the Epstein Frame, which is a device to characterize some physical properties of magnetic materials.

Based on the work of Khansari-Zadeh & Billard [4], we apply the proposed "para-model" framework in order to control the trajectory of a dynamical system-based robot, for which we aim to optimize the dynamical performances.

2 General Principle

We consider a nonlinear SISO dynamical system to control:

$$u \mapsto y, \quad \begin{cases} \dot{x} = f_{nl}(x, u) \\ y = Cx \end{cases} \quad (1)$$

where f_{nl} is a nonlinear system, the para-model agent is an application $(y^*, y) \mapsto u$ whose purpose is to control the output y of (1) following an output reference y^* . In simulation, the system 1 is controlled in its "original formulation" without any modification / linearization.

2.1 Definition of the closed-loop

Consider the control scheme depicted in Fig. 1 where C_π is the proposed PMA controller.

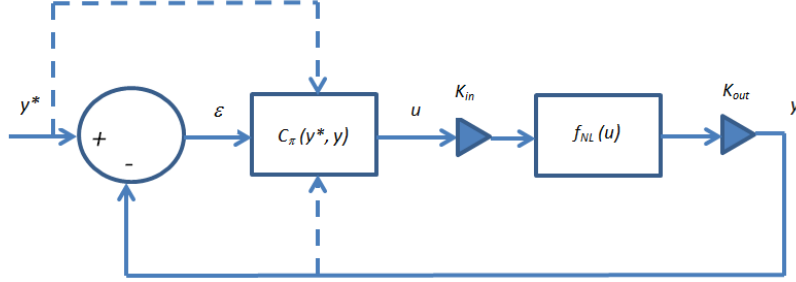


Figure 1: Proposed PMA scheme to control a nonlinear system.

2.2 Definition of the PMA algorithm

For any discrete moment t_k , $k \in \mathbb{N}^*$, one defines the discrete controller C_π such that symbolically:

$$C_\pi : (y, y^*) \mapsto u_k = \int_0^t K_i \varepsilon_{k-1} d\tau \Big|_{k-1}^{\mathbb{R}^2 \rightarrow \mathbb{R}} \underbrace{\{u_{k-1}^i + K_p(k_\alpha e^{-k_\beta k} - y_{k-1})\}}_{u_k^i} \quad (2)$$

where: y^* is the output reference trajectory; K_p and K_i are real positive tuning gains; $\varepsilon_{k-1} = y_{k-1}^* - y_{k-1}$ is the tracking error; $k_\alpha e^{-k_\beta k}$ is an initialization function where k_α and k_β are real constants; practically, the integral part is discretized using e.g. Riemann sums. We define the set of C_π -parameters of the controller as the set of coefficients $\{K_p, K_i, k_\alpha, k_\beta\}$. The internal recursion on u_k^i is defined such as: $u_k^i = u_{k-1}^i + K_p(k_\alpha e^{-k_\beta k} - y_{k-1})$.

3 Application to robot point-to-point movements

3.1 Controllable autonomous dynamical systems

Robot discrete motions are modeled by autonomous Dynamical Systems (DS) that describe the behavior expected by the robot to perform tasks [4]. Consider a state variable $\boldsymbol{\xi} \in \mathbb{R}^d$ that can be used to unambiguously define a discrete motion of a robotic system (e.g. $\boldsymbol{\xi}$ could be a robot's joint angles, position and/or orientation of an arm's end-effector in the operational space, etc.) and define the controllable function \hat{f} such that:

$$\dot{\boldsymbol{\xi}} = f(\boldsymbol{\xi}) = \hat{f}(\boldsymbol{\xi}) + \mathbf{u}, \quad \mathbb{R}^d \mapsto \mathbb{R}^d \quad (3)$$

where \mathbf{u} is the input that allows controlling the model; $f(\boldsymbol{\xi})$ is a continuous function that codes an *exact* specific behavior of the robot and $\hat{f}(\boldsymbol{\xi})$ is the estimated function, derived from $f(\boldsymbol{\xi})$ ¹ that needs to be controlled in order to *comply* with the expected behavior. The expression (3) is integrated using an Euler forward method². Denote $\boldsymbol{\xi}_0 = \mathbf{0}$ the initial configuration and $\boldsymbol{\xi}_f$, the final point that must be reached by the controlled DS.

The purpose is to control (3) by the para-model law (2) in order to maintain $\boldsymbol{\xi}$ "as close as possible" to a trajectory reference $\boldsymbol{\xi}^$ according to the time.*

3.2 Implementation of the \mathcal{C}_π -controller

A possible control scheme is to consider controlling the trajectory $\boldsymbol{\xi}$ that must remain "as close as possible" to the reference $\boldsymbol{\xi}^*$. Therefore, $\boldsymbol{\xi}$ is physically measured and the position of the robot is driven by the \mathcal{C}_π -controller. We build a closed-loop that creates a feedback between (2) and (3). We have "symbolically":

$$\left\{ \begin{array}{l} \mathbf{u}_k = \int_0^t K_i(\boldsymbol{\xi}_{k-1}^* - \boldsymbol{\xi}_{k-1}) d\tau \Big|_{k-1} \underbrace{\{\mathbf{u}_{k-1}^i + K_p(k_\alpha e^{-k_\beta k} - \boldsymbol{\xi}_{k-1})\}}_{\mathbf{u}_k^i} \\ \dot{\boldsymbol{\xi}}_k = \hat{f}(\boldsymbol{\xi}_k) + \mathbf{u}_k \end{array} \right. \quad (4)$$

¹An estimate of $f(\boldsymbol{\xi})$ is built from a set of N demonstrations using any of the state-of-the-art regression methods (see [4]).

²The following standard scheme is used: $\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + h\dot{\boldsymbol{\xi}}$ (where $\dot{\boldsymbol{\xi}}$ is deduced from the estimated function $\hat{f}(\boldsymbol{\xi})$) but when applying (2) to close the loop (4), little oscillations of the trajectory (which remain to study / explain) appear but the closed-loop remains globally "dynamically" stable. To cancel these oscillations, we notice that if one considers a μ factor such as: $\boldsymbol{\xi}_{k+1} = \mu\boldsymbol{\xi}_k + h\dot{\boldsymbol{\xi}}$, $\mu \in [0, 1]$, the modified Euler scheme allows having very nice dynamical performances in closed-loop despite an open-loop trajectory that does not correspond to the original one (Fig. 2) due to the presence of the "disturbing" μ factor inside the Euler scheme.

3.3 Results

3.3.1 Lyapunov-based dynamical performances

Figure 2 presents the trajectory of the robot in open-loop i.e. described exclusively by $\hat{f}(\xi)$; it shows that the state ξ converges to a point that is pretty far from the expected final point ξ_f . Figure 3 presents the trajectory of the robot driven by the Lyapunov function approach [4] that reaches the final point ξ_f .

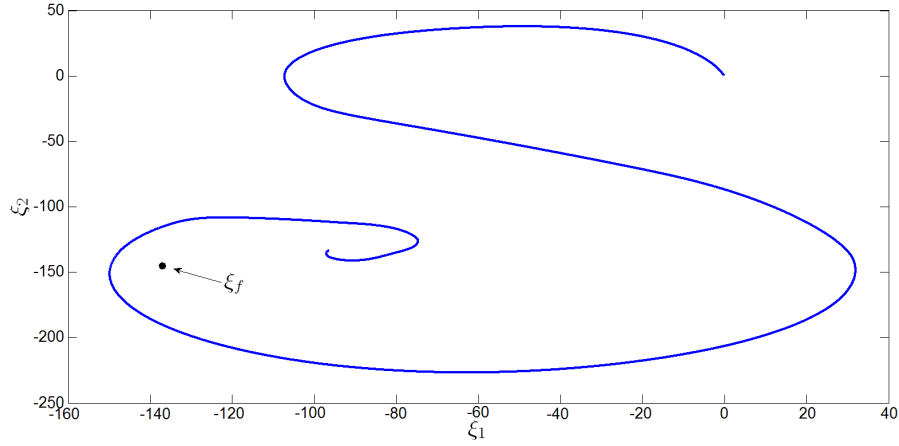


Figure 2: Evolution of the trajectory ξ in open-loop ($\mathbf{u} = \mathbf{0}$).

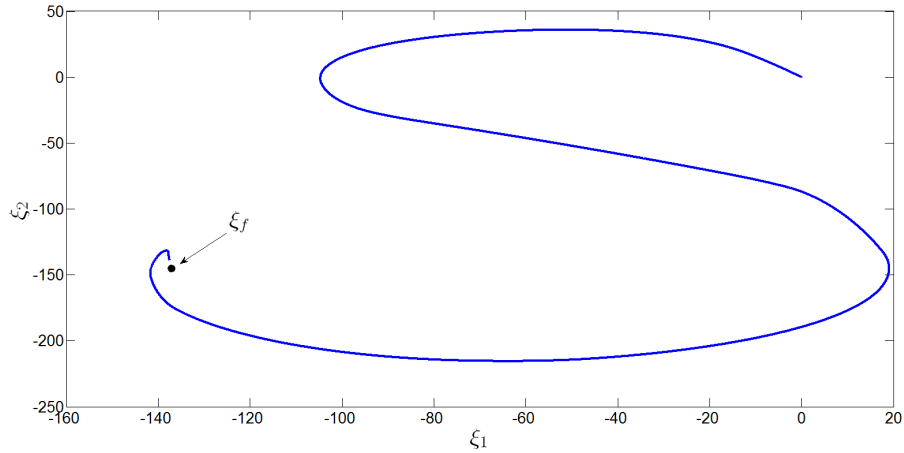


Figure 3: Evolution of the trajectory ξ controlled with a Lyapunov approach.

Figure 4 shows the controlled trajectory ξ by the proposed para-model control according to the time for a particular reference "1" and Fig. 5 shows the same

result in the phase space.

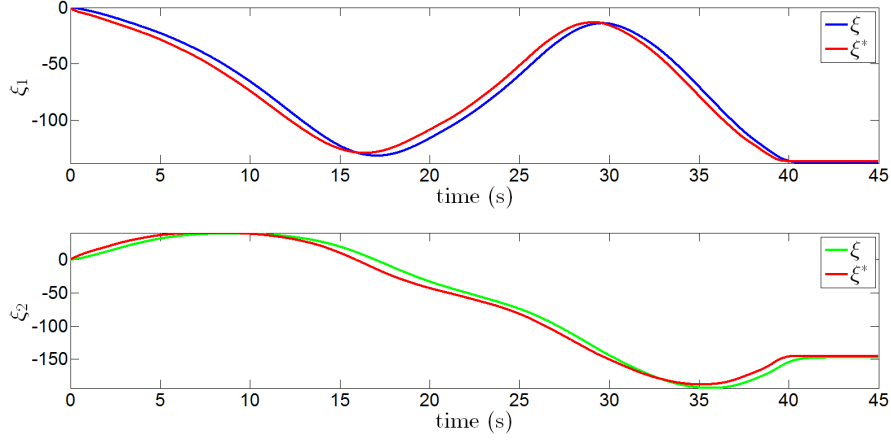


Figure 4: Evolution of the trajectory ξ with respect to the reference ξ^* ("1") according to the time.

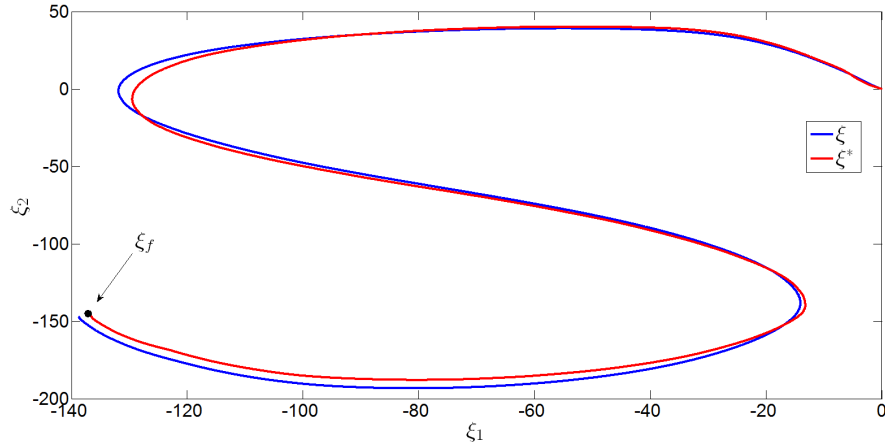


Figure 5: Evolution of the trajectory ξ with respect to the reference ξ^* ("1") in the phase space.

Figure 6 shows the controlled trajectory ξ by the proposed para-model control according to the time for a particular reference "2" and Fig. 7 shows the same result in the phase space. According to the gained experience, the parameters $\{K_p, K_i, k_\alpha, k_\beta\}$ of the para-model law (2) are very flexible and might give interesting dynamical performances in closed-loop even if they have been roughly tuned.

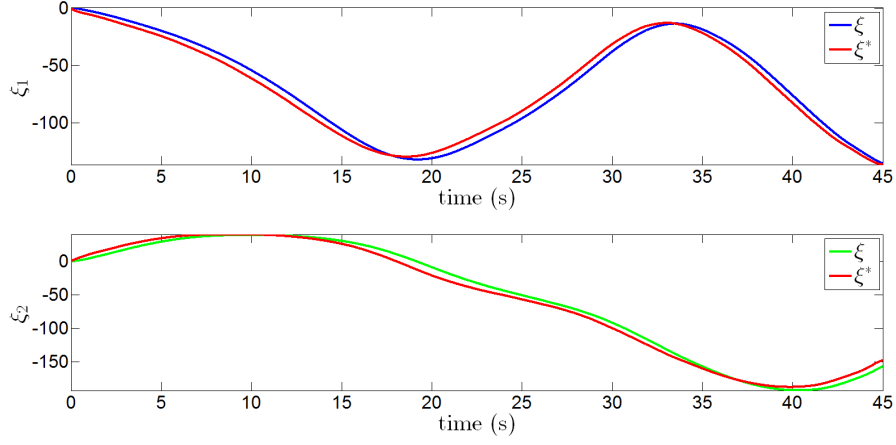


Figure 6: Evolution of the trajectory ξ with respect to the reference ξ^* ("2") according to the time.

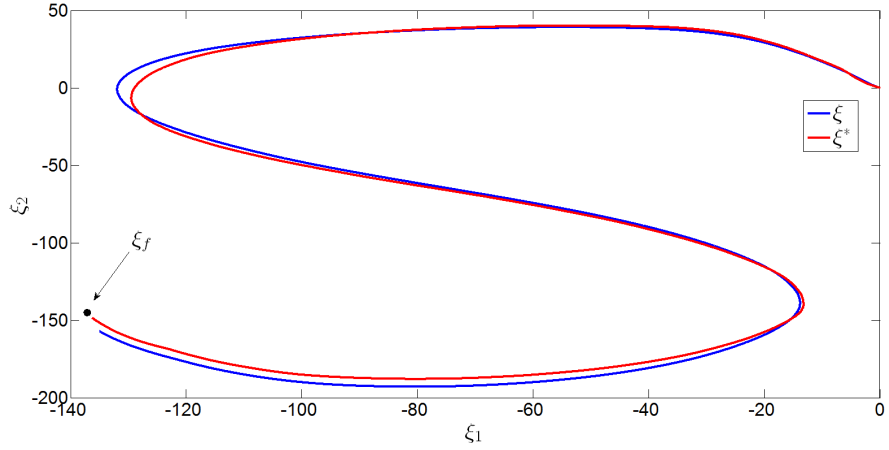


Figure 7: Evolution of the trajectory ξ with respect to the reference ξ^* ("2") in the phase space.

3.3.2 Optimized dynamical performances

To improve the dynamical performances of the closed-loop, we want to solve the problem of finding the most appropriate set of \mathcal{C}_π -parameters relating to the minimization of the ISE (integral square error) performance index such that:

$$\min_{K_p, K_i, k_\alpha, k_\beta} \int_0^{t_f} (\xi - \xi^*)^2 dt$$

where t_f is the final time of the simulation. We are interested in using the "Brute Force Optimization" (BFO) solver [5] that is very convenient and efficient to use. Figures 8 and 9 show the BFO-optimized controlled trajectory ξ by the proposed para-model control according to the time for respectively the references "2" and "1".

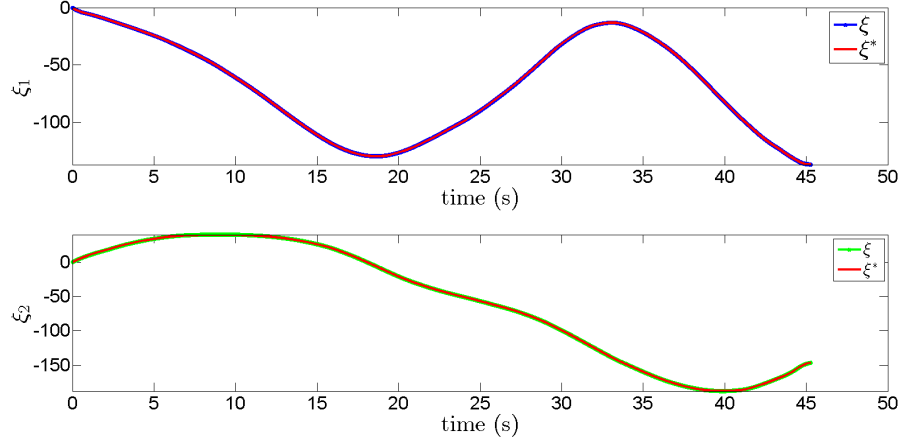


Figure 8: Evolution of the optimized trajectory ξ with respect to the reference ξ^* ("2") according to the time.

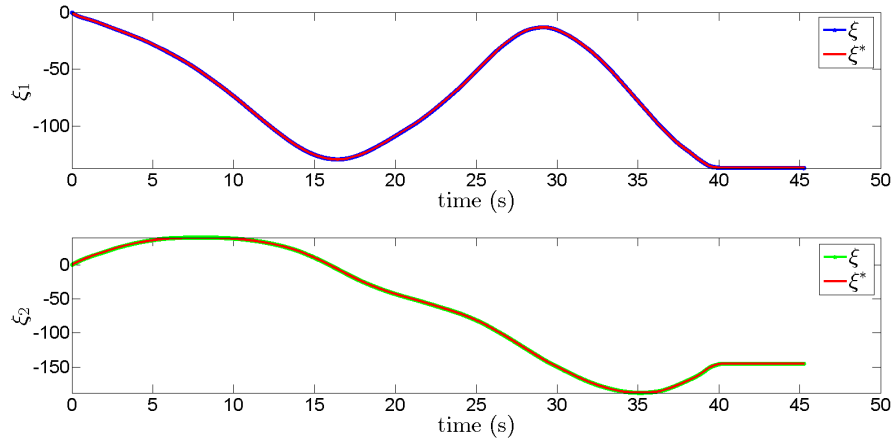


Figure 9: Evolution of the optimized trajectory ξ with respect to the reference ξ^* ("1") according to the time.

3.4 Controlled trajectory with external disturbances

To evaluate the disturbance rejection of the \mathcal{C}_π -controller, we consider adding an external "force" \mathbf{u}^{dist} in (3) such as:

$$\dot{\boldsymbol{\xi}} = f(\boldsymbol{\xi}) = \hat{f}(\boldsymbol{\xi}) + \mathbf{u} + \mathbf{u}^{dist} \quad (5)$$

The following examples illustrate the behavior of the controlled trajectory considering two cases of increasing disturbances: a linear-type disturbance (Fig. 10) and a logarithmic-type disturbance (Fig. 11).

3.4.1 Examples

We consider applying a disturbance \mathbf{u}_k^{dist} over a small period $[t^\alpha, t^\beta] = [1.74, 1.81]$.

case 1 :

$$\begin{cases} \mathbf{u}_k^{dist} = 0.1 + \mathbf{u}_{k-1}^{dist} & \text{when } t_\alpha < t_k < t_\beta \\ \mathbf{u}_k^{dist} = 0 & \text{when } t_k < t_\alpha \text{ and } t_k > t_\beta \end{cases} \quad \text{with } \mathbf{u}_0^{dist} = 0.1 \quad (6)$$

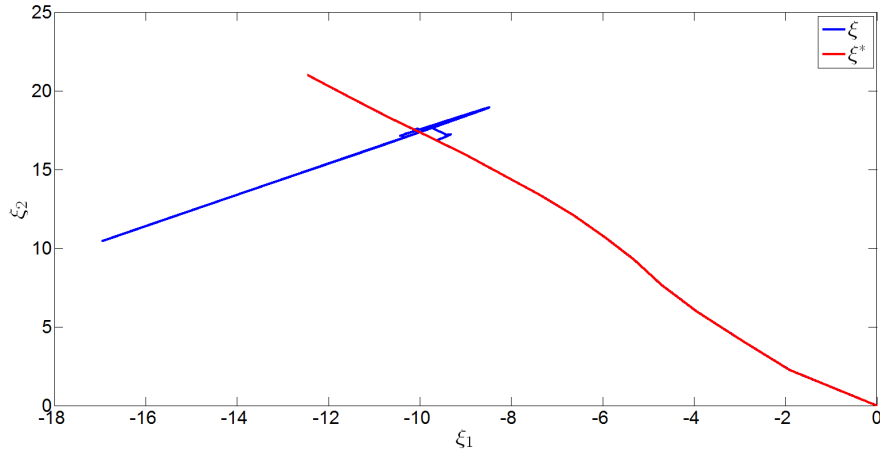


Figure 10: Evolution of the disturbed controlled trajectory considering a linear disturbance (the disturbance is completely rejected after t_β).

case 2 :

$$\begin{cases} \mathbf{u}_k^{dist} = \ln(\mathbf{u}_{k-1}^{dist}) & \text{when } t_\alpha < t_k < t_\beta \\ \mathbf{u}_k^{dist} = 0 & \text{when } t_k < t_\alpha \text{ and } t_k > t_\beta \end{cases} \quad \text{with } \mathbf{u}_0^{dist} = 1.1 \quad (7)$$

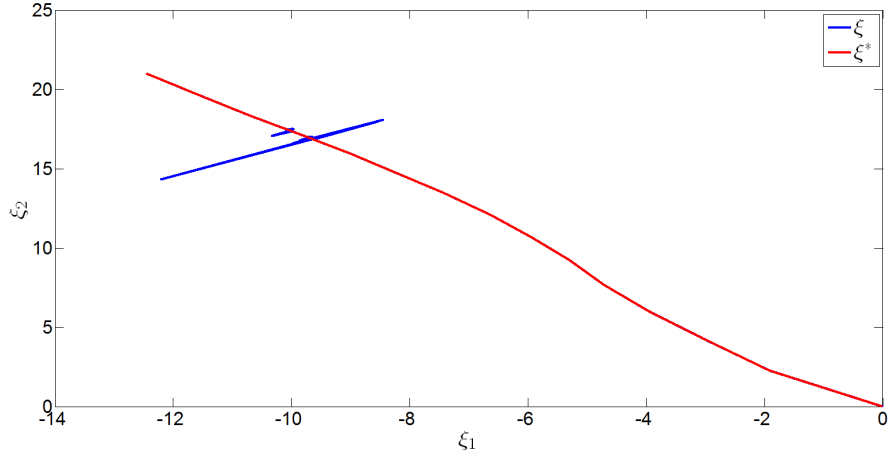


Figure 11: Evolution of the disturbed controlled trajectory considering a logarithmic disturbance (the disturbance is completely rejected after t_β).

Acknowledgement

The author is sincerely grateful to Dr. Edouard Thomas for his strong guidance and his valuable comments that improved this paper.

References

- [1] M. Fliess and C. Join, "Model-free control", International Journal of Control, vol. 86, issue 12, pp. 2228-2252, Jul. 2013 (available at <http://arxiv.org/pdf/1305.7085.pdf>).
- [2] K.J. Åström and P.R. Kumar, "Control: A perspective", Automatica, vol. 50, issue 1, pp. 3-43, Jan. 2014 (available at <http://www.sciencedirect.com/science/article/pii/S0005109813005037>).
- [3] L. Michel, "A para-model agent for dynamical systems", preprint arXiv, Oct. 2016 (available at <http://arxiv.org/abs/1202.4707>).
- [4] S.M. Khansari-Zadeh and A. Billard, "Learning Control Lyapunov Function to Ensure Stability of Dynamical System-based Robot Reaching Motions". Robotics and Autonomous Systems, vol. 62, no 6, pp. 752-765, 2014 (available at http://lasa.epfl.ch/publications/uploadedFiles/Khansari_Billard_RAS2014.pdf).
- [5] M. Porcelli and Ph. L. Toint, BFO, "A trainable derivative-free Brute Force Optimizer for nonlinear bound-constrained optimization and equilibrium computations with continuous and discrete variables", *Namur Center*

for Complex Systems, In Support, Tech. Reports, no. naXys-06-2015, Belgium, Jul. 2015 (available at http://www.optimization-online.org/DB_FILE/2015/07/4986.pdf).